

Sub
AI
Sub
BI

1 1. A computer implemented process for managing
2 exceptions throwable during execution of methods in one or
3 more classes by a machine, each method including an
4 exception handler array defining exception handlers
5 associated with the method, the process comprising:
6 combining the exception handler arrays for two or
7 more methods into a single exception handler table.

1 2. The process of claim 1 including combining all
2 exception handler arrays for all methods in a class in the
3 single exception handler table.

1 3. The process of claim 1 including combining all
2 exception handler arrays for all methods in all classes in
3 the single exception handler table.

1 4. The process of claim 1 including combining all
2 exception handler arrays for all methods in a Java package
3 in the single exception handler table.

1 5. The process of claim 1 where a method is included in
2 a class file and the step of combining all exception handler
3 arrays includes combining the exception handler arrays for
4 all methods in a class file in the single exception handler
5 table.

1 6. The process of claim 1 further including
2 searching the exception handler table when an
3 exception is thrown while executing one of the methods
4 including locating a first matching exception in the single
5 exception handler table.

1 7. The process of claim 6 where the searching step
2 includes retrieving in order exception handler entries from
3 the exception handler table and checking the type and range
4 of each exception handler for the first matching exception
5 handler.

1 8. The process of claim 7 further comprising stopping
2 searching if a current exception handler does not match and
3 is the last handler for the top most level of protected code
4 in an associated method.

1 9. The process of claim 1 where the class files are
2 Java class files.

1 10. The process of claim 1 where the methods in one or
2 more classes are grouped in a package where the package
3 includes a package data structure including first and second
4 portions, the process including storing the exception
5 handler table in the first portion of the package and all
6 methods in the second portion of the package.

1 11. The process of claim 10 where the step of combining
2 includes concatenating the exception handler arrays
3 including loading each exception handler array into the
4 first portion of the package data structure in accordance
5 with a predefined ordering.

1 12. The process of claim 11 where the predefined
2 ordering is determined based on the ordering of methods
3 stored in the second portion of the package data structure.

Sub A2
1 13. The process of claim 1 where the machine is a
2 virtual machine implemented on a resource-constrained
3 device.

Sub B1
1 14. The process of claim 13 where the resource-
2 constrained device is a smart card.

1 15. The process of claim 14 where the methods in one or
2 more classes are grouped in a package and the package is
3 installed on the smart card.

1 16. The process of claim 15 further including creating a
2 package where the package includes a package data structure
3 including first and second portions, the process including
4 concatenating the exception handler arrays for each of the
5 methods into a exception handler table, storing the
6 exception handler table in the first portion of the package
7 and all methods in the second portion of the package.

Sub A2
1 17. A method minimizing the amount of storage required
2 for a runtime stack when executing a program, the runtime
3 stack maintained at runtime during the execution of the
4 program by a machine for storing one or more frames where
5 each frame includes a return pointer to an invoking method
6 that called a currently executing method in the program, the
7 method comprising:

8 combining exception handler information for methods
9 included in the program into a combined exception handler
10 table; and

11 locating and searching the combined exception
12 handler table when an exception is thrown during execution
13 of one of the methods to locate the exception handler
14 information without requiring the storage on the runtime

15 stack of a pointer to the exception handler information.

1 18. The method of claim 17 where the pointer is a direct
2 pointer to the exception handler information.

Sub B 1 19. The method of claim 17 where the program is a Java
2 program.

Sub AA 1 20. The method of claim 19 where the machine is a
2 virtual machine implementing a Java™ virtual machine.

1 21. The method of claim 17 where the program includes a
2 package of methods, the methods in one or more classes, and
3 where the virtual machine is implemented in a resource
4 constrained device on which the package is installed and
5 executing.

1 22. The method of claim 21 where the resource
2 constrained device is a smart card.

1 23. The method of claim 21 further including registering
2 the package in a registry service at installation, the
3 registry service maintaining a pointer and a range, the
4 pointer indicating a location in the resource constrained
5 device of the combined exception handler table associated
6 with a given package, the range defining a range of
7 addresses in the resource constrained device at which
8 methods associated with the package are located.

1 24. The method of claim 23 where the step of locating
2 includes locating a package associated with a currently
3 executing method including comparing an address at which an
4 exception was thrown against the range for each package

5 registered in the registry service, the searching step
6 including searching the combined exception handler table
7 associated with a located package.

1 25. A method of converting class files into a converted
2 applet for execution on a resource constrained device
3 including;

4 receiving one or more class files, each class file
5 including one or more methods, each method including an
6 exception handler array defining exception handlers
7 catchable by the method;

8 defining a data structure for storing the methods
9 and exception handlers for the converted applet including a
10 first and second portion;

11 defining an ordering for the methods and loading the
12 methods according to the ordering in the second portion of
13 the data structure;

14 combining the exception handler arrays for all
15 methods in a single exception handler table including
16 ordering the exception handler arrays according to the
17 ordering defined for the methods and storing the single
18 exception handler array in the first portion of the data
19 structure.

Sub
AS
1 26. A computer implemented process for managing
2 exceptions throwable during execution of two or more methods
3 in one or more classes by a virtual machine, each method
4 included in a class and including an exception handler array
5 defining exception handlers associated with the method, the
6 individual exception handler arrays combined and forming a
7 single exception handler table for the two or more methods,
8 the process comprising:

9 searching the exception handler table when an

10 exception is thrown while executing one of the methods
11 including locating a first matching exception in the single
12 exception handler table.

1 27. A computer-implemented system for managing
2 exceptions throwable during execution of methods in one or
3 more classes by a machine, each method including an
4 exception handler array defining exception handlers
5 associated with the method, the system comprising
6 instructions to:
7 combine the exception handler arrays for all methods
8 into a single exception handler table.

1 28. A computer-implemented system for minimizing the
2 amount of storage required for a runtime stack when
3 executing a program, the runtime stack maintained at runtime
4 during the execution of the program by a machine for storing
5 one or more frames where each frame includes a return
6 pointer to an invoking method that called a currently
7 executing method in the program, the system comprising
8 instructions to:
9 combine the exception handler information for two or
10 more methods included in the program into a combined
11 exception handler table; and
12 locate and search the combined exception handler
13 table when an exception is thrown during execution of one of
14 the methods to locate the exception handler information
15 without requiring the storage on the runtime stack of a
16 pointer to the exception handler information.

1 29. A computer-implemented system for converting class
2 files into a converted applet for execution on a resource
3 constrained device, the system comprising instructions to:

4 receive one or more class files, each class file
5 including one or more methods, each method including an
6 exception handler array defining exception handlers
7 catchable by the method;
8 define a data structure for storing the methods and
9 exception handlers for the converted applet including a
10 first and second portion;
11 define an ordering for the methods and loading the
12 methods according to the ordering in the second portion of
13 the data structure;
14 combine the exception handler arrays for all methods
15 in a single exception handler table including order the
16 exception handler arrays according to the order defined for
17 the methods and store the single exception handler array in
18 the first portion of the data structure.

1 30. A computer-implemented system for managing
2 exceptions throwable during execution of methods in one or
3 more classes by a virtual machine, each method in a class
4 described by a class file and including an exception handler
5 array defining exception handlers associated with the
6 method, the individual exception handler arrays combined and
7 forming a single exception handler table for two or more
8 methods, the system comprising instructions to:
9 search the exception handler table when an exception
10 is thrown while executing one of the two or more methods
11 including locate a first matching exception in the single
12 exception handler table.

ADD AS >